



# Worst-case global optimization of black-box functions through Kriging and relaxation

Julien Marzat, Eric Walter, H  l  ne Piet-Lahanier

## ► To cite this version:

Julien Marzat, Eric Walter, H  l  ne Piet-Lahanier. Worst-case global optimization of black-box functions through Kriging and relaxation. Journal of Global Optimization, 2013, 55 (4), pp.707-727. 10.1007/s10898-012-9899-y . hal-00682475

**HAL Id: hal-00682475**

**<https://hal.science/hal-00682475>**

Submitted on 6 Apr 2012

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destin  e au d  p  t et    la diffusion de documents scientifiques de niveau recherche, publi  s ou non,   manant des   tablissements d'enseignement et de recherche fran  ais ou   trangers, des laboratoires publics ou priv  s.

# Worst-case global optimization of black-box functions through Kriging and relaxation

Julien Marzat · Eric Walter ·  
Hélène Piet-Lahanier

Received: date / Accepted: date

**Abstract** A new algorithm is proposed to deal with the worst-case optimization of black-box functions evaluated through costly computer simulations. The input variables of these *computer experiments* are assumed to be of two types. *Control variables* must be tuned while *environmental variables* have an undesirable effect, to which the design of the control variables should be robust. The algorithm to be proposed searches for a minimax solution, i.e., values of the control variables that minimize the maximum of the objective function with respect to the environmental variables. The problem is particularly difficult when the control and environmental variables live in continuous spaces. Combining a relaxation procedure with Kriging-based optimization makes it possible to deal with the continuity of the variables and the fact that no analytical expression of the objective function is available in most real-case problems. Numerical experiments are conducted to assess the accuracy and efficiency of the algorithm, both on analytical test functions with known results and on an engineering application.

**Keywords** computer experiments · continuous minimax · efficient global optimization · expected improvement · fault diagnosis · Kriging · robust optimization · worst-case analysis

## 1 Introduction

Computer models of complex processes are now extensively used in all domains of pure and applied sciences. These models can be viewed as black-box functions that provide a response to sampled input values. Choosing where to sample the input space can be viewed as the design of *computer experiments* [1]. The optimal

---

J. Marzat · H. Piet-Lahanier  
ONERA – The French Aerospace Lab, F-91761 Palaiseau, France  
Tel.: +33 1 80 38 66 50  
E-mail: julien.marzat@onera.fr, helene.piet-lahanier@onera.fr

J. Marzat · E. Walter  
L2S, CNRS-SUPELEC-Univ Paris-Sud, 91192 Gif-sur-Yvette, France  
E-mail: eric.walter@lss.supelec.fr

sampling obviously depends on the goal of the computer experiments. We assume here that this goal is the optimal choice of *control variables*. In most real-life problems, the objective function has no closed-form expression and is expensive to evaluate. In this context, surrogate models such as those provided by the response-surface methodology, Kriging, radial basis functions, splines or neural networks are widely used [2–4]. The idea is to substitute the evaluation of some simple function for the costly simulation of the complex computer model.

Much research has been carried out on how to choose sampling points in the input space of the control variables on which the fitting of the surrogate model should be achieved. Space-filling sampling heavily suffers from the curse of dimensionality [5]. A more interesting strategy explores new input values sequentially according to some sampling criterion motivated by the improvement of an estimate of a global optimizer. Kriging [6], also known as Gaussian-process regression [7], is especially relevant in this context. Under clearly defined assumptions, it provides the best linear unbiased prediction on the continuous space of inputs, as well as a measure of the uncertainty of this prediction. These elements have been exploited to build a highly popular sampling criterion known as *Expected Improvement* (EI) and the *Efficient Global Optimization* (EGO) algorithm [8], which allows the optimization of black-box functions on a very reduced sampling budget compared to other strategies [9]. Variations around EGO can be found in [10–12], and convergence results in [13]. Many successful applications of EGO in engineering have been reported, e.g., in [11, 14–17]. Kriging and EGO will serve as a basis for the study presented here.

In many real-life applications, the sole consideration of control variables corresponds to a simplistic version of the problem, as *environmental variables* that affect performance should also be taken into account. For instance, a control law or an estimation filter for an aeronautical vehicle are subject to measurement noise, strong uncertainty on the model parameters, variations of the atmospheric conditions (temperature, pressure) and wind turbulence. In such a case, one is looking for a design of the control variables that is robust to the effect of these environmental variables. When dealing with such a robust design, a probabilistic or deterministic point of view may be adopted [18]. In the probabilistic framework, a distribution of the environmental variables is assumed, and performance is assessed by the expected value of some robustness measure. However, a design that is good on average may prove to be poor for particular values of the environmental variables. In the deterministic framework, it is assumed that the environmental variables belong to some known compact set and performance is assessed by the worst possible value of some robustness measure. The design that is best in the worst case is obviously conservative on average, and the choice between the probabilistic and deterministic points of view should be made on a case-by-case basis.

In the probabilistic context, some papers have addressed robust Kriging-based optimization with respect to environmental variables. In [19–21], Monte-Carlo simulations are performed for each sampled value of a space-filling design of the control variables and a Kriging model is fitted on the resulting mean and variance, before achieving optimization by classical algorithms. In [22–24], the EI criterion has been extended to take into account a probability distribution for the environmental variables. The underlying idea is to minimize a weighted average of the response over a discrete set of values for the environmental variables.

In the worst-case context, few algorithms seem to have been reported yet to deal with environmental variables for the robust optimization of black-box functions evaluated by costly computer experiments. Most of the techniques available use evolutionary algorithms, which are usually computationally expensive [25, 26], and thus impractical in a context of costly simulations. An interesting attempt combining a surrogate model with evolutionary optimization has been reported in [27].

The present paper presents an alternative algorithm for the continuous minimax optimization of black-box functions. We propose to rely on the iterative relaxation procedure proposed in [28] and to combine it with Kriging-based optimization. Relaxation makes it possible to take into account continuous infinite spaces for both the control and environmental variables, unlike the discrete probabilistic formulation used in previous work. Kriging-based optimization may also drastically reduce computational load, compared to evolutionary-based strategies.

This paper is organized as follows. Section 2 formulates the optimization problem under consideration and briefly presents numerical minimax optimization. Section 3 recalls elements about Gaussian Processes and Kriging-based optimization. The new algorithm combining Kriging-based optimization and a relaxation procedure is presented in Section 4. Section 5 demonstrates its efficiency first on test functions from [29] and [25–27], then on a simplified version of an actual engineering problem.

## 2 Numerical minimax optimization

Denote the vector of control variables by  $\mathbf{x}_c$ , the vector of environmental variables by  $\mathbf{x}_e$  and the corresponding scalar value of the objective function as computed by the complex model by  $y(\mathbf{x}_c, \mathbf{x}_e)$ . Assume that  $\mathbf{x}_c \in \mathbb{X}_c$  and  $\mathbf{x}_e \in \mathbb{X}_e$  where  $\mathbb{X}_c$  and  $\mathbb{X}_e$  are known compact sets. The aim of minimax optimization is to find  $\hat{\mathbf{x}}_c$  and  $\hat{\mathbf{x}}_e$  such that

$$\{\hat{\mathbf{x}}_c, \hat{\mathbf{x}}_e\} = \arg \min_{\mathbf{x}_c \in \mathbb{X}_c} \max_{\mathbf{x}_e \in \mathbb{X}_e} y(\mathbf{x}_c, \mathbf{x}_e). \quad (1)$$

This is especially relevant if the design must remain valid when the worst environmental disturbance occurs. Minimax ideas are widespread in many areas, e.g., mechanical design [30], control [31, 32] or fault diagnosis [33, 34]. This kind of problem is also closely related to bi-level optimization [35], robust optimization [36] and game theory [37].

While a considerable amount of work has been devoted to the theory of minimax optimization, relatively few *numerical* algorithms have been proposed to address continuous minimax problems. See [29, 38] for a survey and [39, 40] for recent developments. All of these strategies assume that an analytical expression of the function to be optimized is available, as well as gradient or sub-gradient information. As a result, they are not applicable to the problem considered here. A simple idea would be to find a minimizer  $\hat{\mathbf{x}}_c$  of  $y$  on  $\mathbb{X}_c$  for a fixed value  $\mathbf{x}_e \in \mathbb{X}_e$ , then to maximize  $y$  with respect to  $\mathbf{x}_e$  on  $\mathbb{X}_e$  for this fixed value  $\hat{\mathbf{x}}_c$ , and to alternate these steps. However, the convergence of this algorithm, known as *Best Replay* [41], is not guaranteed and it turns out very often to cycle through useless values of candidate solutions.

To overcome these drawbacks, Shimizu and Aiyoshi [28, 42] have proposed to transform the initial problem (1) into the equivalent problem

$$\begin{cases} \min_{\mathbf{x}_c \in \mathbb{X}_c} \tau, \\ \text{subject to } y(\mathbf{x}_c, \mathbf{x}_e) \leq \tau, \forall \mathbf{x}_e \in \mathbb{X}_e. \end{cases} \quad (2)$$

This problem has an infinite number of constraints and is therefore still intractable. An iterative procedure can however be employed to find an approximate solution, by relaxing the problem (2) into

$$\begin{cases} \min_{\mathbf{x}_c \in \mathbb{X}_c} \tau, \\ \text{subject to } y(\mathbf{x}_c, \mathbf{x}_e) \leq \tau, \forall \mathbf{x}_e \in \mathcal{R}_e, \end{cases} \quad (3)$$

where  $\mathcal{R}_e$  is a *finite* set containing values of  $\mathbf{x}_e$  that have already been explored. Algorithm 1 summarizes this strategy.

---

**Algorithm 1** Minimax optimization via relaxation

---

1: Pick  $\mathbf{x}_e^{(1)} \in \mathbb{X}_e$  and set  $\mathcal{R}_e = \{\mathbf{x}_e^{(1)}\}$  and  $i = 1$ .

2: Compute

$$\mathbf{x}_c^{(i)} = \arg \min_{\mathbf{x}_c \in \mathbb{X}_c} \left\{ \max_{\mathbf{x}_e \in \mathcal{R}_e} y(\mathbf{x}_c, \mathbf{x}_e) \right\}$$

3: Compute

$$\mathbf{x}_e^{(i+1)} = \arg \max_{\mathbf{x}_e \in \mathbb{X}_e} y(\mathbf{x}_c^{(i)}, \mathbf{x}_e)$$

4: If

$$y(\mathbf{x}_c^{(i)}, \mathbf{x}_e^{(i+1)}) - \max_{\mathbf{x}_e \in \mathcal{R}_e} y(\mathbf{x}_c^{(i)}, \mathbf{x}_e) < \varepsilon_R$$

then return  $\{\mathbf{x}_c^{(i)}, \mathbf{x}_e^{(i+1)}\}$  as an approximate solution to the initial minimax problem (1).

Else, append  $\mathbf{x}_e^{(i+1)}$  to  $\mathcal{R}_e$ , increment  $i$  by 1 and go to Step 2.

---

The threshold  $\varepsilon_R$  specifies the accuracy of the desired solution. Note that if the procedure is stopped before the termination condition of Step 4 is reached, then an approximate solution is still obtained, corresponding to a higher threshold  $\varepsilon'_R$ . This is particularly interesting when the number of evaluations of  $y$  is restricted. Under reasonable assumptions, the main loop has been proven to terminate after a finite number of iterations [28].

This algorithm, also used in [39], is generic and leaves open the choice of the optimization procedures to be employed at Steps 2 and 3. Kriging-based optimization seems particularly appropriate in a context of costly evaluations.

### 3 Kriging-based optimization

Throughout this section, some black-box function  $f(\boldsymbol{\xi})$  is assumed to depend on a  $d$ -dimensional vector of inputs  $\boldsymbol{\xi} \in \mathbb{X} \subset \mathbb{R}^d$ . In the context of the relaxation procedure,  $\boldsymbol{\xi}$  will either consist of control variables or of environmental variables, depending of the optimization step, and  $f(\cdot)$  will be  $y(\cdot, \mathbf{x}_e)$  at Step 2 and  $-y(\mathbf{x}_c^{(i)}, \cdot)$  at Step 3. The aim is to find a global minimizer  $\hat{\boldsymbol{\xi}}$  of  $f(\cdot)$  on the continuous bounded space  $\mathbb{X}$

$$\hat{\boldsymbol{\xi}} = \arg \min_{\boldsymbol{\xi} \in \mathbb{X}} f(\boldsymbol{\xi}), \quad (4)$$

with as few evaluations as possible. The objective function  $f(\cdot)$  can only be evaluated at sampled values of  $\boldsymbol{\xi}$  and no gradient information is assumed to be available. An interesting tool in this context is Kriging.

#### 3.1 Kriging

Kriging [6] models the black-box objective function  $f(\cdot)$  as a Gaussian process (GP). A GP can be seen as the generalization of finite-space Gaussian distributions to a function space of infinite dimension. Just as a Gaussian distribution is fully specified by its mean and covariance matrix, a GP is characterized by its mean and covariance *functions* [7, 43]. In what follows, the objective function  $f(\cdot)$  will be modeled as a GP

$$F(\boldsymbol{\xi}) = \mathbf{p}^T(\boldsymbol{\xi})\mathbf{b} + Z(\boldsymbol{\xi}). \quad (5)$$

The mean function is  $m_F(\boldsymbol{\xi}) = \mathbf{p}^T(\boldsymbol{\xi})\mathbf{b}$ , where  $\mathbf{p}(\boldsymbol{\xi})$  is some known vector of regressors (usually chosen constant or polynomial in  $\boldsymbol{\xi}$ ) and  $\mathbf{b}$  is a vector of unknown regression coefficients to be estimated.  $Z(\boldsymbol{\xi})$  is a zero-mean GP with covariance function  $\text{cov}(\cdot, \cdot)$ , usually expressed as

$$\text{cov}(Z(\boldsymbol{\xi}_i), Z(\boldsymbol{\xi}_j)) = \sigma_Z^2 R(\boldsymbol{\xi}_i, \boldsymbol{\xi}_j), \quad (6)$$

where  $\sigma_Z^2$  is the process variance and  $R(\cdot, \cdot)$  a parametric correlation function. The parameters of  $R(\cdot, \cdot)$  and  $\sigma_Z^2$  must be chosen *a priori* or estimated from the available data. Many choices are possible for  $R(\cdot, \cdot)$ , one of the most frequent being the *power exponential correlation function* [1],

$$R(\boldsymbol{\xi}_i, \boldsymbol{\xi}_j) = \exp \left( - \sum_{k=1}^d \left| \frac{\xi_i(k) - \xi_j(k)}{\theta_k} \right|^{p_k} \right), \quad (7)$$

with  $\xi_i(k)$  the  $k$ -th component of  $\boldsymbol{\xi}_i$ . The parameters  $\theta_k > 0$  quantify how the influence of data points decreases with their distance to the point of prediction. In this paper, all  $p_k$  are chosen equal to 2, which corresponds to a smooth prediction. Note that  $R(\boldsymbol{\xi}_i, \boldsymbol{\xi}_j) \rightarrow 1$  when  $\|\boldsymbol{\xi}_i - \boldsymbol{\xi}_j\| \rightarrow 0$  and  $R(\boldsymbol{\xi}_i, \boldsymbol{\xi}_j) \rightarrow 0$  when  $\|\boldsymbol{\xi}_i - \boldsymbol{\xi}_j\| \rightarrow \infty$ . In this paper, we use *empirical Kriging* where covariance parameters are estimated from the data by maximum likelihood.

It is assumed that a set of training data  $\mathbf{f}_n = [f(\boldsymbol{\xi}_1), \dots, f(\boldsymbol{\xi}_n)]^T$  has already been computed, corresponding to an initial sampling  $\mathcal{X}_n = [\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_n]$  of  $n$  points in  $\mathbb{X}$ . The Kriging predictor is the best linear unbiased predictor (BLUP) of  $f(\boldsymbol{\xi})$ ,

for any  $\boldsymbol{\xi} \in \mathbb{X}$ . It consists of two parts [44]. The first one is the prediction of the mean of the Gaussian process at  $\boldsymbol{\xi} \in \mathbb{X}$  by

$$\hat{f}(\boldsymbol{\xi}) = \mathbf{p}^T(\boldsymbol{\xi}) \hat{\mathbf{b}} + \mathbf{r}(\boldsymbol{\xi})^T \mathbf{R}^{-1} (\mathbf{f}_n - \mathbf{P} \hat{\mathbf{b}}), \quad (8)$$

where  $\mathbf{R}$  is the  $n \times n$  matrix such that

$$\mathbf{R}_{i,j} = R(\boldsymbol{\xi}_i, \boldsymbol{\xi}_j), \quad (9)$$

$\mathbf{r}(\boldsymbol{\xi})$  is the  $n$  vector

$$\mathbf{r}(\boldsymbol{\xi}) = [R(\boldsymbol{\xi}_1, \boldsymbol{\xi}), \dots, R(\boldsymbol{\xi}_n, \boldsymbol{\xi})]^T, \quad (10)$$

$\mathbf{P}$  is the  $n \times \dim \mathbf{b}$  matrix

$$\mathbf{P} = [\mathbf{p}(\boldsymbol{\xi}_1), \dots, \mathbf{p}(\boldsymbol{\xi}_n)]^T, \quad (11)$$

and  $\hat{\mathbf{b}}$  is the maximum-likelihood estimate of  $\mathbf{b}$  from the available data  $\{\mathcal{X}_n; \mathbf{f}_n\}$

$$\hat{\mathbf{b}} = \left( \mathbf{P}^T \mathbf{R}^{-1} \mathbf{P} \right)^{-1} \mathbf{P}^T \mathbf{R}^{-1} \mathbf{f}_n. \quad (12)$$

The prediction of the mean of the GP is linear in  $\mathbf{f}_n$  and interpolates the training data as  $\hat{f}(\boldsymbol{\xi}_i) = f(\boldsymbol{\xi}_i)$  for  $i = 1, \dots, n$ . Kriging can also be seen as a linear predictor with a weighted sum on a basis of functions [7].

A second, very important part of the Kriging prediction is the estimate of the *variance of the prediction error*

$$\hat{\sigma}^2(\boldsymbol{\xi}) = \sigma_Z^2 \left( 1 - \mathbf{r}(\boldsymbol{\xi})^T \mathbf{R}^{-1} \mathbf{r}(\boldsymbol{\xi}) \right), \quad (13)$$

which quantifies the accuracy of the Kriging prediction at  $\boldsymbol{\xi}$ . It is small near already sampled data points (even zero at their exact location), and large far from them [11]. This has been used for the definition of optimization algorithms that look for promising sampling points  $\boldsymbol{\xi}$  in the sense that either  $\hat{f}(\boldsymbol{\xi})$  is small or  $\hat{\sigma}^2(\boldsymbol{\xi})$  is large.

### 3.2 Optimization

Kriging-based optimization [8] iteratively samples new points where  $f(\cdot)$  should be evaluated, to improve the estimate of a global optimizer. Sampling is made according to a criterion  $J(\cdot)$  that measures the interest of an additional evaluation at  $\boldsymbol{\xi}$ , given the available data, the Kriging predictor (8) and the corresponding uncertainty measure (13),

$$\boldsymbol{\xi}_{n+1} = \arg \max_{\boldsymbol{\xi} \in \mathbb{X}} J(\boldsymbol{\xi}, \mathcal{X}_n, \mathbf{f}_n, \hat{f}(\boldsymbol{\xi}), \hat{\sigma}(\boldsymbol{\xi})). \quad (14)$$

A common choice for  $J(\cdot)$  is *Expected Improvement* (EI) [8], defined as

$$\text{EI}(\boldsymbol{\xi}, f_{\min}^n, \hat{f}, \hat{\sigma}) = \hat{\sigma}(\boldsymbol{\xi}) [u \Phi(u) + \phi(u)], \quad (15)$$

where  $\Phi$  is the cumulative distribution function and  $\phi$  the probability density function of the normalized Gaussian distribution  $\mathcal{N}(0, 1)$ , with

$$u = \frac{f_{\min}^n - \hat{f}(\boldsymbol{\xi})}{\hat{\sigma}(\boldsymbol{\xi})} \quad (16)$$

and

$$f_{\min}^n = \min_{i=1 \dots n} \{f(\boldsymbol{\xi}_i)\}. \quad (17)$$

Maximizing EI achieves a trade-off between local search (numerator of  $u$ ) and the exploration of unknown areas (where  $\hat{\sigma}$  is high), which is appropriate in a context of global optimization. It is at the core of the Efficient Global Optimization (EGO) algorithm, described in Algorithm 2. For the sake of simplicity, the expression of  $\text{EI}(\boldsymbol{\xi}, f_{\min}^n, \hat{f}, \hat{\sigma})$  is contracted into  $\text{EI}(\boldsymbol{\xi})$  in the description of the algorithms.

---

**Algorithm 2** Efficient Global Optimization

---

- 1: Choose an initial sampling  $\mathcal{X}_n = \{\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_n\}$  in  $\mathbb{X}$
  - 2: Compute  $\mathbf{f}_n = \{f(\boldsymbol{\xi}_1), \dots, f(\boldsymbol{\xi}_n)\}$
  - 3: **while**  $\max_{\boldsymbol{\xi} \in \mathbb{X}} \text{EI}(\boldsymbol{\xi}) > \varepsilon_{\text{EI}}$  **and**  $n < n_{\max}$  **do**
  - 4:   Fit the Kriging model on the known data points  $\{\mathcal{X}_n, \mathbf{f}_n\}$  with (8)-(13)
  - 5:   Find  $f_{\min}^n = \min_{i=1 \dots n} \{f(\boldsymbol{\xi}_i)\}$
  - 6:   Find  $\boldsymbol{\xi}_{n+1} = \arg \max_{\boldsymbol{\xi} \in \mathbb{X}} \text{EI}(\boldsymbol{\xi})$
  - 7:   Compute  $f(\boldsymbol{\xi}_{n+1})$ , append it to  $\mathbf{f}_n$  and append  $\boldsymbol{\xi}_{n+1}$  to  $\mathcal{X}_n$
  - 8:    $n \leftarrow n + 1$
  - 9: **end while**
- 

The initial sampling at Step 1 can be performed by, e.g., Latin Hypercube Sampling (LHS) or any other space-filling design [43]. A rule of thumb is to take ten samples per dimension of input space, so  $n = 10d$  [8]. At Step 6, a new point where to sample the function is searched for by maximizing EI (15). The EI (15) has a closed-form expression and can easily be computed and differentiated at any point of  $\mathbb{X}$ . Its evaluation only involves computation on the surrogate model. Following Sasena's work [11], we used the DIRECT algorithm [45], but other implementations could be used as well. The procedure is repeated until one of the stopping criteria is met, either the exhaustion of the sampling budget  $n_{\max}$  or the reaching of the threshold  $\varepsilon_{\text{EI}}$  on EI.

#### 4 Worst-case global optimization of black-box functions

This section addresses the initial minimax problem (1) for black-box functions by combining EGO (Algorithm 2) with the relaxation procedure (Algorithm 1). At Steps 2 and 3 of Algorithm 1, two functions depending on a vector of input variables should be optimized. Two independent EGO algorithms can be used for this purpose, requiring two initial samplings,  $\mathcal{X}_0^c$  on  $\mathbb{X}_c$  and  $\mathcal{X}_0^e$  on  $\mathbb{X}_e$ . The



complete procedure is detailed in Algorithm 3 and called MiMaReK for MiniMax optimization via Relaxation and Kriging. In this algorithm, the index  $i$  is the number of elements in  $\mathcal{R}_e$ , while  $j$  is the number of iterations. Note that Steps 1 to 4 of MiMaReK correspond exactly to the four steps of Algorithm 1. At Step 1a, the initial vector  $\mathbf{x}_e^{(1)}$  may be set arbitrarily.

At Step 2 the aim is to find a minimizer  $\mathbf{x}_c^{(i)}$  of the function  $\max_{\mathbf{x}_e \in \mathcal{R}_e} y(\mathbf{x}_c, \mathbf{x}_e)$ , where  $\mathcal{R}_e$  consists of a finite number of already explored values of  $\mathbf{x}_e$ . The computation of  $\mathbf{y}_j^c$  at Step 2b is thus carried out by picking, for each point  $\mathbf{x}_c$  of the current design  $\mathcal{X}_j^c$ , the empirical maximum of  $y(\mathbf{x}_c, \mathbf{x}_e)$  over all elements  $\mathbf{x}_e$  of  $\mathcal{R}_e$ . This requires  $j \times i$  evaluations of  $y$  at each iteration of the while loop of Step 2. We chose to rely on the same initial sampling  $\mathcal{X}_0^c$  at each iteration of the relaxation. This reduces the computational cost to  $i$  evaluations of  $y(\cdot, \cdot)$  per iteration. It should also be noted that the fitting of the Kriging model and maximization of EI are performed on  $\max_{\mathbf{x}_e \in \mathcal{R}_e} y(\mathbf{x}_c, \mathbf{x}_e)$ , which is a function of  $\mathbf{x}_c$  only, instead of  $y$ .

At Step 3 the function to be maximized is  $y(\mathbf{x}_c^{(i)}, \mathbf{x}_e)$ , EGO is thus employed to minimize  $-y(\mathbf{x}_c^{(i)}, \mathbf{x}_e)$  with  $\mathbf{x}_c^{(i)}$  the fixed value obtained at Step 2. This is a function of  $\mathbf{x}_e$  only. The same initial sampling  $\mathcal{X}_0^e$  may also be used at each call of this step, however this does not reduce the overall computational cost significantly. Most optimization tasks in Steps 2 and 3 simply require picking the optimum in a finite set of values. Only Steps 2(c)iii and 3(c)iii require the use of an optimization algorithm for the simple-to-evaluate EI function (15), as explained in Section 3.2. To use Algorithm 3, one needs to set seven parameters. The dimensions of the initial samplings  $n_c$  and  $n_e$  may be fixed respectively at  $10 \dim \mathbf{x}_c$  and  $10 \dim \mathbf{x}_e$ . The maximal number of iterations allowed per EGO algorithm  $n_{EI}^c$  and  $n_{EI}^e$  depend on the computational resources available and the time taken by the evaluation of the objective function  $y(\cdot, \cdot)$ . The tolerance parameter  $\varepsilon_R$  on the stopping condition of the relaxation procedure determines the accuracy of the minimax optimum. Tolerances  $\varepsilon_{EI}^c$  and  $\varepsilon_{EI}^e$  on the values of EI for each of the EGO algorithms determine the accuracy with which the intermediate optimization tasks are carried out. Empirical considerations on how to choose these values are given in Section 5.1.3.

## 5 Examples of application

Two types of applications are developed to assess the efficiency of the minimax strategy. In Sections 5.1 and 5.2, test functions with known results serve as references to study the behavior of the procedure. In Section 5.3, Algorithm 3 is used to address the robust tuning of hyperparameters of a fault diagnosis method for an aeronautical system.

### 5.1 Benchmark 1 (Rustem and Howe [29])

Seven convex-concave test functions for minimax optimization have been defined in Chapter 5 of [29]. The dimensions of these problems range from 2 control variables and 2 environmental variables, up to 5 control variables and 5 environmental variables. In the aforementioned book, three descent methods (Kiwiel's and two

**Algorithm 3** MiMaReK, MiniMax optimization via Relaxation and Kriging

---

Set  $\varepsilon_R, \varepsilon_{EI}^c, n_{EI}^c, \varepsilon_{EI}^e, n_{EI}^e, n_c, n_e$ .

---

**1. Step 1**

- (a) Choose randomly  $\mathbf{x}_e^{(1)}$  in  $\mathbb{X}_e$ . Initialize  $\mathcal{R}_e = \{\mathbf{x}_e^{(1)}\}$ . Set  $i \leftarrow 1$ .
- (b) Choose a design  $\mathcal{X}_0^c = \{\mathbf{x}_{c,1}, \dots, \mathbf{x}_{c,n_c}\}$  by LHS in  $\mathbb{X}_c$ .
- (c) Choose a design  $\mathcal{X}_0^e = \{\mathbf{x}_{e,1}, \dots, \mathbf{x}_{e,n_e}\}$  by LHS in  $\mathbb{X}_e$ .

**while**  $e > \varepsilon_R$ **2. Step 2**

- (a) Initialize  $j \leftarrow n_c$  and  $\mathcal{X}_j^c = \mathcal{X}_0^c$ .
- (b) Compute  $\mathbf{y}_j^c = \left\{ \max_{\mathbf{x}_e \in \mathcal{R}_e} \{y(\mathbf{x}_{c,1}, \mathbf{x}_e)\}, \dots, \max_{\mathbf{x}_e \in \mathcal{R}_e} \{y(\mathbf{x}_{c,n_c}, \mathbf{x}_e)\} \right\}$ .
- (c) **while**  $\max_{\mathbf{x}_c \in \mathbb{X}_c} \{EI(\mathbf{x}_c)\} > \varepsilon_{EI}^c$  **and**  $j < n_{EI}^c$ 
  - i. Fit a Kriging model on the known data points  $\{\mathcal{X}_j^c, \mathbf{y}_j^c\}$ .
  - ii. Find  $y_{\min}^j = \min_{1 \dots j} \{\mathbf{y}_j^c\}$ .
  - iii. Find the next point of interest  $\mathbf{x}_{c,j+1}$  by maximizing  $EI(\mathbf{x}_c)$ .
  - iv. Append  $\mathbf{x}_{c,j+1}$  to  $\mathcal{X}_j^c$ .
  - v. Find  $\max_{\mathbf{x}_e \in \mathcal{R}_e} \{y(\mathbf{x}_{c,j+1}, \mathbf{x}_e)\}$  and append it to  $\mathbf{y}_j^c$ .
  - vi.  $j \leftarrow j + 1$ .
- end while**
- (d) Find  $\mathbf{x}_c^{(i)} = \arg \min_{\mathbf{x}_c \in \mathcal{X}_j^c} \{\mathbf{y}_j^c\}$
- (e) Compute  $e_{\text{prec}} = \max_{\mathbf{x}_e \in \mathcal{R}_e} y(\mathbf{x}_c^{(i)}, \mathbf{x}_e)$

**3. Step 3**

- (a) Initialize  $k \leftarrow n_e$  and  $\mathcal{X}_k^e = \mathcal{X}_0^e$ .
- (b) Compute  $\mathbf{y}_k^e = \left\{ -y(\mathbf{x}_c^{(i)}, \mathbf{x}_{e,1}), \dots, -y(\mathbf{x}_c^{(i)}, \mathbf{x}_{e,n_e}) \right\}$ .
- (c) **while**  $\max_{\mathbf{x}_e \in \mathbb{X}_e} \{EI(\mathbf{x}_e)\} > \varepsilon_{EI}^e$  **and**  $k < n_{EI}^e$ 
  - i. Fit a Kriging model on the known data points  $\{\mathcal{X}_k^e, \mathbf{y}_k^e\}$ .
  - ii. Find  $y_{\max}^k = \min_{1 \dots k} \{\mathbf{y}_k^e\}$ .
  - iii. Find the next point of interest  $\mathbf{x}_{e,k+1}$  by maximizing  $EI(\mathbf{x}_e)$ .
  - iv. Append  $\mathbf{x}_{e,k+1}$  to  $\mathcal{X}_k^e$ .
  - v. Compute  $-y(\mathbf{x}_c^{(i)}, \mathbf{x}_{e,k+1})$  and append it to  $\mathbf{y}_k^e$ .
  - vi.  $k \leftarrow k + 1$ .
- end while**
- (d) Find  $\mathbf{x}_e^{(i+1)} = \arg \min_{\mathbf{x}_e \in \mathcal{X}_k^e} \{\mathbf{y}_k^e\}$  and append it to  $\mathcal{R}_e$

**4. Step 4**

- (a) Compute  $e = y(\mathbf{x}_c^{(i)}, \mathbf{x}_e^{(i+1)}) - e_{\text{prec}}$
- (b)  $i \leftarrow i + 1$

**end while**

types of quasi-Newton schemes) have been compared on these test functions and provided similar results. These results will serve as reference solutions. In what follows, these seven test functions are taken as *black-box* objective functions, and MiMaReK (Algorithm 3) is applied to evaluate the minimax solution. Its results are then compared with the references.

### 5.1.1 Test functions for Benchmark 1

The  $i$ -th component of the vector of control variables  $\mathbf{x}_c$  is denoted by  $x_{ci}$ , and the  $i$ -th component of the vector of environmental variables  $\mathbf{x}_e$  by  $x_{ei}$ . The analytical expressions of the seven test functions are

$$f_1(\mathbf{x}_c, \mathbf{x}_e) = 5(x_{c1}^2 + x_{c2}^2) - (x_{e1}^2 + x_{e2}^2) + x_{c1}(-x_{e1} + x_{e2} + 5) + x_{c2}(x_{e1} - x_{e2} + 3), \quad (18)$$

$$f_2(\mathbf{x}_c, \mathbf{x}_e) = 4(x_{c1} - 2)^2 - 2x_{e1}^2 + x_{c1}^2 x_{e1} - x_{e2}^2 + 2x_{c2}^2 x_{e2}, \quad (19)$$

$$f_3(\mathbf{x}_c, \mathbf{x}_e) = x_{c1}^4 x_{e2} + 2x_{c1}^3 x_{e1} - x_{c2}^2 x_{e2}(x_{e2} - 3) - 2x_{c2}(x_{e1} - 3)^2, \quad (20)$$

$$f_4(\mathbf{x}_c, \mathbf{x}_e) = -\sum_{i=1}^3 (x_{ei} - 1)^2 + \sum_{i=1}^2 (x_{ci} - 1)^2 + x_{e3}(x_{c2} - 1) + x_{e1}(x_{c1} - 1) + x_{e2}x_{c1}x_{c2}, \quad (21)$$

$$f_5(\mathbf{x}_c, \mathbf{x}_e) = -(x_{c1} - 1)x_{e1} - (x_{c2} - 2)x_{e2} - (x_{c3} - 1)x_{e3} + 2x_{c1}^2 + 3x_{c2}^2 + x_{c3}^2 - x_{e1}^2 - x_{e2}^2 - x_{e3}^2, \quad (22)$$

$$f_6(\mathbf{x}_c, \mathbf{x}_e) = x_{e1}(x_{c1}^2 - x_{c2} + x_{c3} - x_{c4} + 2) + x_{e2}(-x_{c1} + 2x_{c2}^2 - x_{c3}^2 + 2x_{c4} + 1) \quad (23)$$

$$+ x_{e3}(2x_{c1} - x_{c2} + 2x_{c3} - x_{c4}^2 + 5) + 5x_{c1}^2 + 4x_{c2}^2 + 3x_{c3}^2 + 2x_{c4}^2 - \sum_{i=1}^3 x_{ei}^2,$$

$$f_7(\mathbf{x}_c, \mathbf{x}_e) = 2x_{c1}x_{c5} + 3x_{c4}x_{c2} + x_{c5}x_{c3} + 5x_{c4}^2 + 5x_{c5}^2 - x_{c4}(x_{e4} - x_{e5} - 5) \quad (24)$$

$$+ x_{c5}(x_{e4} - x_{e5} + 3) + \sum_{i=1}^3 (x_{ei}(x_{ci}^2 - 1)) - \sum_{i=1}^5 x_{ei}^2.$$

For each of the test functions of Benchmark 1, Table 1 gives the continuous bounded search sets  $\mathbb{X}_c$  and  $\mathbb{X}_e$  for the control and environmental variables, the minimax solution  $(\hat{\mathbf{x}}_c, \hat{\mathbf{x}}_e)$  and the corresponding value of the objective function, as obtained in [29].  $\mathbb{X}_c$  is unbounded in [29], so it was chosen large enough to contain the reference solution and the initialization points of the descent algorithms.

Table 1: Reference solutions for the seven minimax test functions

Test function	$\mathbb{X}_c$	$\mathbb{X}_e$	Reference $\hat{\mathbf{x}}_c$	Reference $\hat{\mathbf{x}}_e$	$f_i(\hat{\mathbf{x}}_c, \hat{\mathbf{x}}_e)$
$f_1(\mathbf{x}_c, \mathbf{x}_e)$	$[-5; 5]^2$	$[-5; 5]^2$	$-0.4833$ $-0.3167$	$0.0833$ $-0.0833$	$-1.6833$
$f_2(\mathbf{x}_c, \mathbf{x}_e)$	$[-5; 5]^2$	$[-5; 5]^2$	$1.6954$ $-0.0032$	$0.7186$ $-0.0001$	$1.4039$
$f_3(\mathbf{x}_c, \mathbf{x}_e)$	$[-5; 5]^2$	$[-3; 3]^2$	$-1.1807$ $0.9128$	$2.0985$ $2.666$	$-2.4688$
$f_4(\mathbf{x}_c, \mathbf{x}_e)$	$[-5; 5]^2$	$[-3; 3]^3$	$0.4181$ $0.4181$	$0.709$ $1.0874$ $0.709$	$-0.1348$
$f_5(\mathbf{x}_c, \mathbf{x}_e)$	$[-5; 5]^3$	$[-1; 1]^3$	$0.1111$ $0.1538$ $0.2$	$0.4444$ $0.9231$ $0.4$	$1.345^1$
$f_6(\mathbf{x}_c, \mathbf{x}_e)$	$[-5; 5]^4$	$[-2; 2]^3$	$-0.2316$ $0.2228$ $-0.6755$ $-0.0838$	$0.6195$ $0.3535$ $1.478$	$4.543$
$f_7(\mathbf{x}_c, \mathbf{x}_e)$	$[-5; 5]^5$	$[-3; 3]^5$	$1.4252$ $1.6612$ $1.2585$ $-0.9744$ $-0.7348$	$0.5156$ $0.8798$ $0.2919$ $0.1198$ $-0.1198$	$-6.3509$

### 5.1.2 Results with the new minimax algorithm on Benchmark 1

The results to be presented have been obtained with the following tuning parameters,  $\varepsilon_R = 10^{-3}$ ,  $\varepsilon_{EI}^c = \varepsilon_{EI}^e = 10^{-4}$ ,  $n_{EI}^c = n_{EI}^e = 100$ . One hundred random initializations of the procedure have been carried out to assess repeatability. The corresponding average results are given in Table 2. They have been compared to the reference values from Table 1 by computing the absolute value of their relative deviation, for the minimizer  $\hat{\mathbf{x}}_c$ , the maximizer  $\hat{\mathbf{x}}_e$  and the function value  $f_i(\hat{\mathbf{x}}_c, \hat{\mathbf{x}}_e)$ . The results of this comparison are reported in Table 3. The worst deviation for the value of the objective function is 0.29%. More dispersion is observed for the values of the minimizers and maximizers, since different values of the arguments can lead to very close values of the objective function. The worst deviation observed is approximately 7%. The number of evaluations required for the black-box functions is divided by  $(\dim \mathbb{X}_c + \dim \mathbb{X}_e)$  (i.e. the total problem dimension) to reveal the intrinsic complexity of each problem. For example, the third test function  $f_3$  requires a high number of evaluations relative to the minimax dimension,

<sup>1</sup> [29] indicated 0.1345, however we obtained 1.345 by evaluating the function at the given references for the minimax solution, so we assumed it was a typo.

Table 2: Average results for 100 runs of the procedure for the test functions

Test function	average $\hat{\mathbf{x}}_c$	average $\hat{\mathbf{x}}_e$	average $f_i(\hat{\mathbf{x}}_c, \hat{\mathbf{x}}_e)$	Std. deviation $f_i$
$f_1(\mathbf{x}_c, \mathbf{x}_e)$	-0.4860 -0.3299	0.0792 -0.0774	-1.6824	0.0295
$f_2(\mathbf{x}_c, \mathbf{x}_e)$	1.6966 -0.0033	0.7206 -0.0156	1.4036	0.0012
$f_3(\mathbf{x}_c, \mathbf{x}_e)$	-1.1818 0.9119	2.0898 2.6859	-2.4689	0.0682
$f_4(\mathbf{x}_c, \mathbf{x}_e)$	0.4181 0.4205	0.6953 1.085 0.6988	-0.1352	0.0213
$f_5(\mathbf{x}_c, \mathbf{x}_e)$	0.1142 0.1559 0.202	0.4415 0.9149 0.3954	1.345	0.006
$f_6(\mathbf{x}_c, \mathbf{x}_e)$	-0.2239 0.2305 -0.6629 -0.0398	0.6037 0.4032 1.4903	4.5485	0.0207
$f_7(\mathbf{x}_c, \mathbf{x}_e)$	1.3922 1.6087 1.1938 -0.9684 -0.7171	0.4719 0.8149 0.2288 0.1086 -0.1337	-6.3334	0.1561

Table 3: Relative deviation of results from reference, in percent

Test function	$\hat{\mathbf{x}}_c$	$\hat{\mathbf{x}}_e$	$f_i(\hat{\mathbf{x}}_c, \hat{\mathbf{x}}_e)$	Evaluations per dimension
$f_1(\mathbf{x}_c, \mathbf{x}_e)$	1.65 %	6.01 %	<b>0.05</b> %	64
$f_2(\mathbf{x}_c, \mathbf{x}_e)$	0.07 %	0.28 %	<b>0.02</b> %	147
$f_3(\mathbf{x}_c, \mathbf{x}_e)$	0.02 %	0.3 %	<b>0.004</b> %	251
$f_4(\mathbf{x}_c, \mathbf{x}_e)$	0.29 %	0.89 %	<b>0.29</b> %	94
$f_5(\mathbf{x}_c, \mathbf{x}_e)$	1.4 %	0.88 %	<b>0.001</b> %	81
$f_6(\mathbf{x}_c, \mathbf{x}_e)$	2.16 %	0.96 %	<b>0.12</b> %	382
$f_7(\mathbf{x}_c, \mathbf{x}_e)$	2.8 %	7.37 %	<b>0.28</b> %	402

even if  $(\dim \mathbb{X}_c + \dim \mathbb{X}_e)$  is only equal to 4. Compare with  $f_5$  that evolves on 6 dimensions, but seems less difficult to optimize.

Figure 1 shows examples of the dispersion of the results for 100 initializations. These zoomed-in views around the reference value should be analyzed while keeping in mind that these functions take large values on their domains of definition; here a rough inner approximation of the domain of variation over  $\mathbb{X}_c \times \mathbb{X}_e$  gives  $[-414, 496]$  for  $f_2$  and  $[-95, 170]$  for  $f_5$ .

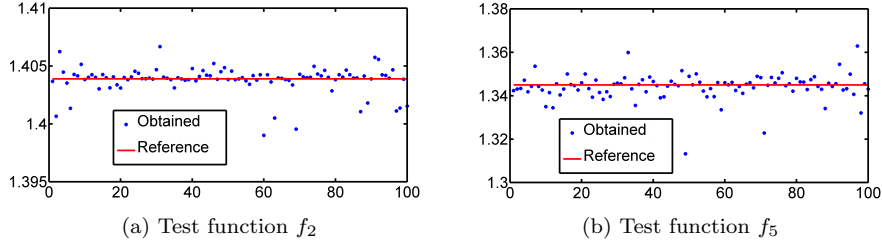


Fig. 1: Estimated minimax values for 100 initializations, compared to reference

### 5.1.3 Remarks on convergence and parameter tuning

As mentioned in Section 2, the relaxation procedure provides a suboptimal minimax solution if interrupted before the threshold  $\varepsilon_R$  is reached. This is illustrated in Figure 2, where values of the current estimates of the minimax solutions at each iteration of the relaxation procedure are shown, along with reference values. Good estimates of the minimax solution turn out to have been obtained well before termination.

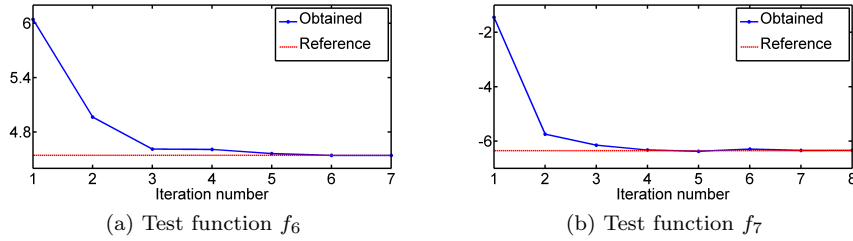


Fig. 2: Estimate of the minimax value of objective functions at each iteration of the relaxation procedure, compared with references

The tolerances on EI and the relaxation procedure were deliberately chosen small in this benchmark, to assess convergence to reference value. However, Figure 2 suggests that much larger values of the tolerance parameters for the relaxation and EGO algorithms may suffice. To check this hypothesis, an empirical

campaign has been conducted for the test function  $f_5$ . Its results are reported in Figure 3, for a grid of thresholds  $(\varepsilon_R, \varepsilon_{EI})$  between  $10^{-1}$  and  $10^{-4}$  (the same  $\varepsilon_{EI}$  has been used for the two EGO algorithms). A trade-off between accuracy (small deviation) and complexity (number of evaluations) clearly appears. For a constant  $\varepsilon_R$ , lowering  $\varepsilon_{EI}$  improves the quality of the estimation of the minimax value. The converse is also true, and in both cases the number of evaluations of the objective function grows with the diminution of the threshold values.

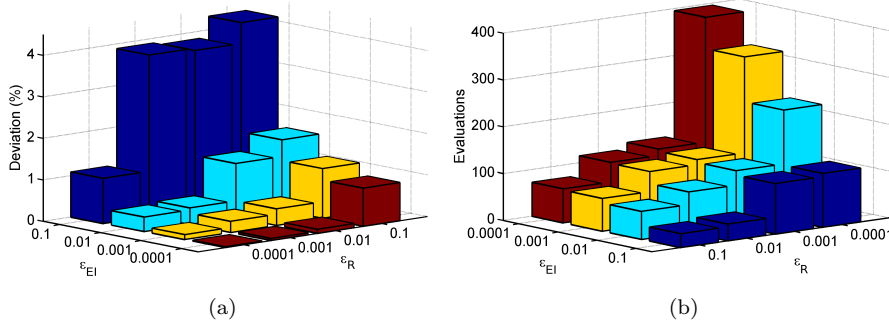


Fig. 3: Deviations from reference (a) and number of evaluations (b) for various choices of the tolerance thresholds  $\varepsilon_R$  and  $\varepsilon_{EI}$  (function  $f_5$ )

## 5.2 Benchmark 2 - computational load comparison

To further assess the computational load of the algorithm, tests have also been carried out with the functions proposed in [25–27], where results with evolutionary algorithms are reported. These test functions are

$$f_8(\mathbf{x}_c, \mathbf{x}_e) = (x_{c1} - 5)^2 - (x_{e1} - 5^2), \quad (25)$$

$$f_9(\mathbf{x}_c, \mathbf{x}_e) = \min\{3 - 0.2x_{c1} + 0.3x_{e1}, 3 + 0.2x_{c1} - 0.1x_{e1}\}, \quad (26)$$

$$f_{10}(\mathbf{x}_c, \mathbf{x}_e) = \frac{\sin(x_{c1} - x_{e1})}{\sqrt{x_{c1}^2 + x_{e1}^2}}, \quad (27)$$

$$f_{11}(\mathbf{x}_c, \mathbf{x}_e) = \frac{\cos\left(\sqrt{x_{c1}^2 + x_{e1}^2}\right)}{\sqrt{x_{c1}^2 + x_{e1}^2} + 10}, \quad (28)$$

$$f_{12}(\mathbf{x}_c, \mathbf{x}_e) = 100(x_{c2} - x_{c1}^2)^2 + (1 - x_{c1})^2 - x_{e1}(x_{c1} + x_{c2}^2) - x_{e2}(x_{c1}^2 + x_{c2}), \quad (29)$$

$$f_{13}(\mathbf{x}_c, \mathbf{x}_e) = (x_{c1} - 2)^2 + (x_{c2} - 1)^2 + x_{e1}(x_{c1}^2 - x_{c2}) + x_{e2}(x_{c1} + x_{c2} - 2). \quad (30)$$

Table 4 summarizes the reference results as indicated in [26]. The numerical results obtained with MiMaReK with the same tuning parameters as in Section 5.1.2, averaged on one hundred random initializations, are reported in Table 5. Very

few evaluations of the black-box functions turn out to be required on these low-dimensional problems. These numbers are also consistent with those obtained on the first benchmark for similar dimensions. It should be noted that in [25,26], between  $10^4$  and  $10^5$  evaluations of the functions were required to achieve similar performance.

Table 4: Reference solutions for the six functions of Benchmark 2

Function	$\mathbb{X}_c$	$\mathbb{X}_e$	Reference $\hat{\mathbf{x}}_c$	Reference $\hat{\mathbf{x}}_e$	$f_i(\hat{\mathbf{x}}_c, \hat{\mathbf{x}}_e)$
$f_8(\mathbf{x}_c, \mathbf{x}_e)$	[0; 10]	[0; 10]	5	5	0
$f_9(\mathbf{x}_c, \mathbf{x}_e)$	[0; 10]	[0; 10]	0	0	3
$f_{10}(\mathbf{x}_c, \mathbf{x}_e)$	[0; 10]	[0; 10]	10	2.1257	$9.7794 \cdot 10^{-2}$
$f_{11}(\mathbf{x}_c, \mathbf{x}_e)$	[0; 10]	[0; 10]	7.0441	10	$4.2488 \cdot 10^{-2}$
$f_{12}(\mathbf{x}_c, \mathbf{x}_e)$	$[-0.5; 0.5] \times [0; 1]$	$[0; 10]^2$	0.5 0.25	0 0	0.25
$f_{13}(\mathbf{x}_c, \mathbf{x}_e)$	$[-1; 3]^2$	$[0; 10]^2$	1 1	any any	1

Table 5: Absolute deviation of results for Benchmark 2

Test function	$\hat{\mathbf{x}}_c$	$\hat{\mathbf{x}}_e$	$f_i(\hat{\mathbf{x}}_c, \hat{\mathbf{x}}_e)$	Evaluations per dimension
$f_8(\mathbf{x}_c, \mathbf{x}_e)$	$6 \cdot 10^{-5}$	$4.7 \cdot 10^{-6}$	<b><math>3.5 \cdot 10^{-9}</math></b>	83
$f_9(\mathbf{x}_c, \mathbf{x}_e)$	$4.5 \cdot 10^{-15}$	$1.4 \cdot 10^{-14}$	<b><math>4.4 \cdot 10^{-16}</math></b>	51
$f_{10}(\mathbf{x}_c, \mathbf{x}_e)$	$1.6 \cdot 10^{-14}$	$8 \cdot 10^{-2}$	<b><math>3.03 \cdot 10^{-7}</math></b>	141
$f_{11}(\mathbf{x}_c, \mathbf{x}_e)$	$3.7 \cdot 10^{-3}$	$5 \cdot 10^{-15}$	<b><math>2.7 \cdot 10^{-5}</math></b>	157
$f_{12}(\mathbf{x}_c, \mathbf{x}_e)$	$2.38 \cdot 10^{-4}$	$1.6 \cdot 10^{-3}$	<b><math>1 \cdot 10^{-3}</math></b>	38
$f_{13}(\mathbf{x}_c, \mathbf{x}_e)$	$6.1 \cdot 10^{-3}$	–	<b><math>4 \cdot 10^{-3}</math></b>	68

### 5.3 Engineering application

In this section, a realistic application of minimax design is considered. The problem under study is the robust tuning of hyperparameters for fault diagnosis methods that monitor the behavior of an aircraft.

#### 5.3.1 Fault diagnosis

One of the most important purposes of fault diagnosis is to detect unexpected changes (*faults*) in a monitored process as early as possible, before they lead to



a complete breakdown. When a dynamical model of the system is available, for instance under the form of a set of differential equations, it can be used to predict its expected behavior and compare it to the one actually observed. This consistency check makes it possible to compute *residuals*, i.e., signals that should remain small as long as there is no fault and become sufficiently large to be noticeable whenever a fault occurs. Here the residual generator is a Luenberger observer [34] that estimates the output of the system based on the knowledge of its model and its input and output. A residual is generated by comparing this estimate with the measured output. This residual is then analyzed by a decision test, which can be a simple threshold or a statistical test, to provide a Boolean decision about whether a fault is present. The decision test considered here is the classical CUSUM test [46]. Figure 4 summarizes the procedure<sup>2</sup>.

The design problem to be considered now is the choice of the tuning parameters, also called *hyperparameters*, of the Luenberger observer and CUSUM test, with respect to some performance criterion. This design should be robust to environmental variables, such as the amount of measurement noise and the size of the fault to be detected. This can be formulated as minimax optimization, where the control variables are the hyperparameters, the environmental variables describe properties of the disturbances and the objective function is some performance measure of fault diagnosis. Algorithm 3 is called upon to solve this problem.

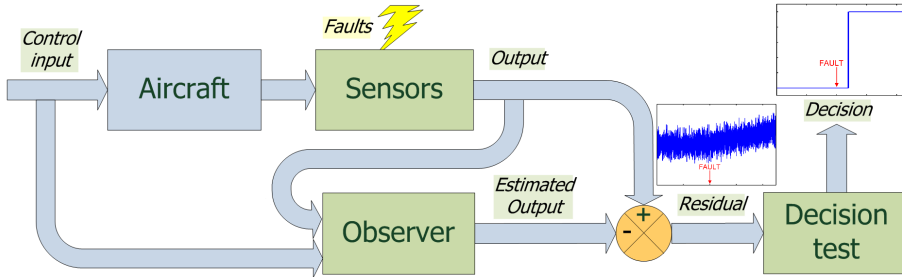


Fig. 4: Model-based fault diagnosis with observer and decision test

### 5.3.2 Test case

The system under study is an aircraft that may suffer a sensor fault, i.e., a persisting error on the measured values provided by sensors. For the sake of simplicity, only the longitudinal motion of the aircraft at a constant altitude of 6000m is considered here. The state of the vehicle is then determined by the values of three variables, namely the angle of attack, angular rate and Mach number, which form the state vector  $\mathbf{x} = [\alpha, q, M]^T$ . The control input is the rudder angle  $\delta$ , and the available measurement provided by an accelerometer is the normal acceleration  $a_z$ . The linearized model around the operating point

<sup>2</sup> Actuators are omitted in this simplified version of the problem

$\mathbf{x}_0 = [\bar{\alpha}, \bar{q}, \bar{M}]^T = [20 \text{ deg}, 18.4 \text{ deg/s}, 3]^T$  obeys the following state-space equation, after discretization by the explicit Euler method with a time step of 0.02s,

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\delta_k \\ a_z = \mathbf{C}\mathbf{x}_k + \mathbf{D}\delta_k + v_k + w_{f,k} \end{cases} \quad (31)$$

where

$$\mathbf{A} = \begin{bmatrix} 0.9163 & 0.0194 & 0.0026 \\ -5.8014 & 0.9412 & 0.5991 \\ -0.0485 & -0.005 & 0.996 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} -0.0279 \\ -2.5585 \\ -0.0019 \end{bmatrix} \quad (32)$$

$$\mathbf{C} = \begin{bmatrix} -2.54 & 0 & -0.26 \end{bmatrix}, \quad \mathbf{D} = -0.204$$

This model is simulated on a time horizon of 50s. A sensor fault  $w_f$  on the measurement of  $a_z$  occurs at time 25s. This fault is simulated by adding a ramp with slope  $\varsigma$  to the current measured value. The measurement noise  $v$  is uniformly distributed in the interval  $[-\zeta, \zeta]$ . The environmental variables  $\zeta$  and  $\varsigma$  form the vector  $\mathbf{x}_e$  to which the tuning should be robust. The set  $\mathbb{X}_e$  is an axis-aligned rectangle such that  $\zeta \in [10^{-7}, 10^{-3}]$  and  $\varsigma \in [10^{-3}, 10^{-1}]$ .

### 5.3.3 Fault diagnosis filter

The empirical mean and variance of the residual obtained by comparing the output predicted by the observer and its measurement are estimated on the first 100 values. The residual is then normalized to zero mean and unit variance according to these estimates, in order to compensate for the differences of behavior induced by a change of values of the environmental variables. Thus, the same tuning of a statistical test is applicable to different levels of noise. A CUSUM test is used on the normalized residual to provide a Boolean decision on whether a fault is present. The response of the observer is governed by three poles  $p_1, p_2$  and  $p_3$  to be placed between 0 (fast response) and 1 (slow response) with no imaginary part to avoid oscillations, and smaller than the real parts of the poles of the system. The CUSUM test has two parameters, namely the size  $\mu$  of the change to be detected and a threshold  $\lambda$ . The method to be tuned has thus five hyperparameters  $\mathbf{x}_c = [p_1, p_2, p_3, \mu, \lambda]^T$ , and  $\mathbb{X}_c$  is assumed to be an axis-aligned box such that  $p_1, p_2, p_3 \in [0; 0.8]$ ,  $\mu \in [0.01; 1]$  and  $\lambda \in [1; 10]$ .

### 5.3.4 Performance indices

A trade-off must be achieved between false-detection and non-detection rates. Figure 5 shows time zones of the Boolean decision function that are used to define performance indices. The value of the function before  $t_{\text{on}}$  and after  $t_{\text{hor}}$  is not to be taken into account, while  $t_{\text{from}}$  is the instant of time at which the fault occurs. The indices that will be used for performance evaluation [47] are

- the *false-detection rate*  $r_{\text{fd}} = (\sum_i t_{\text{fd}}^i) / (t_{\text{from}} - t_{\text{on}})$ , where  $t_{\text{fd}}^i$  is the  $i$ -th period of time between  $t_{\text{on}}$  and  $t_{\text{from}}$  where the decision is *true*;

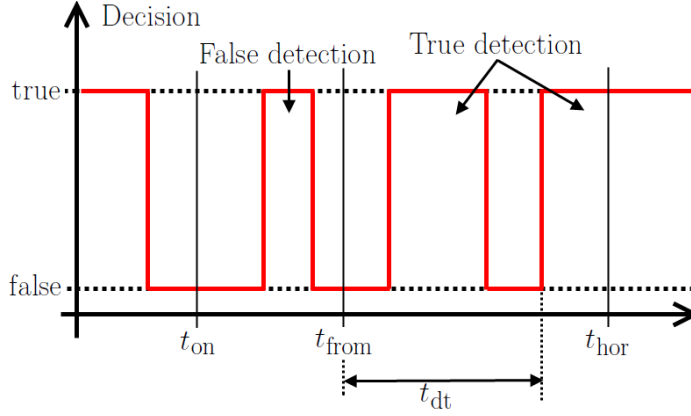


Fig. 5: Time zone parameters for the definition of performance indices

- the *non-detection rate*  $r_{nd} = 1 - r_{td}$ , where  $r_{td} = (\sum_i t_{td}^i) / (t_{hor} - t_{from})$  is the *true-detection rate* with  $t_{td}^i$  the  $i$ -th period of time between  $t_{from}$  and  $t_{hor}$  where the decision is *true*.

The objective function  $y(\mathbf{x}_c, \mathbf{x}_e)$  is  $y = r_{fd} + r_{nd}$ . It achieves a trade-off between the contradictory goals of minimizing false-detection and non-detection, and takes continuous values in  $[0; 2]$ , the best performance corresponding to 0.

### 5.3.5 Results

The parameters of the optimization procedure have been set to  $\varepsilon_R = 10^{-4}$ ,  $n_{EI}^c = n_{EI}^e = 100$ ,  $\varepsilon_{EI}^c = \varepsilon_{EI}^e = 10^{-4}$ . The prior mean of the Gaussian process is assumed constant, while its variance and the parameters  $\theta_k$  of the correlation function (7) are estimated from the available data by maximum likelihood at each iteration of EGO. As in Section 5.1, 100 runs of the minimax procedure have been performed to assess the dispersion of its results. Mean and standard deviation for the best hyperparameters and worst environmental variables, along with corresponding values of the objective function and number of evaluations are reported in Table 6. The dispersion of the hyperparameters obtained suggest that several tunings of the fault diagnosis method allow acceptable performance to be reached. The number of evaluations is relatively low, with an average sampling of approximately 61 points per minimax dimension.

The feasible domain  $\mathbb{X}_e$  for the environmental vector  $\mathbf{x}_e$  is displayed on Figure 6, showing the results for the 100 runs of the procedure on the test case. These results indicate that the worst environmental conditions are located near the smallest value of the fault and highest value of the noise level, which was to be expected in this simple case. Note that on a more complex problem, intuition would not be sufficient to find out what the worst environmental conditions for the best tuning are.

Figure 7 shows the estimated minimax values of typical hyperparameters and the objective function obtained for the 100 random initializations of the procedure. The objective function always takes nonzero values, which indicates that false

Table 6: Results for 100 initializations of the tuning procedure

	Mean	Std. deviation
Best hyperparameter vector $\hat{\mathbf{x}}_c$		
Pole $p_1$	0.7363	$5.2 \cdot 10^{-2}$
Pole $p_2$	0.7058	$6.6 \cdot 10^{-2}$
Pole $p_3$	0.72	$5.3 \cdot 10^{-2}$
Change size $\mu$	0.0714	$4.9 \cdot 10^{-2}$
Threshold $\lambda$	4.5379	0.2
Worst environmental vector $\hat{\mathbf{x}}_e$		
Noise level $\zeta$	$9.3 \cdot 10^{-4}$	$1.1 \cdot 10^{-4}$
Fault slope $\varsigma$	$1.1 \cdot 10^{-3}$	$2 \cdot 10^{-4}$
Minimax cost and number of evaluations per dimension		
Objective function $y$	0.125	$4.7 \cdot 10^{-2}$
Evaluations	61	31

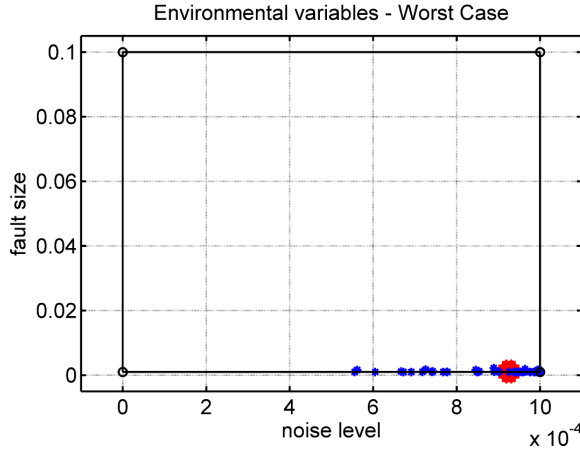


Fig. 6: Worst-case values for the environmental variables, as estimated by Mi-MaReK for 100 replications (small dots); mean value indicated by a large spot and boundaries of  $\mathbb{X}_e$  by a black rectangle

alarms or non detections cannot be avoided. However the worst possible sum of the non-detection and false-alarm rates is evaluated and minimized. With a reverse point of view, one could assess the minimum detectable size of fault for a desired diagnosis performance level.

Figure 8 displays the residual and corresponding Boolean decision obtained via the observer and the CUSUM test tuned at the mean of their optimal values as estimated by the minimax procedure, for the mean of the evaluated worst-case environmental condition. The worst-case residual satisfactorily detects the fault, and no false detection is observed. The detection delay is reasonable, given the

incipient character of the fault and its drowning in high noise. For comparison, a random point is taken in the environmental space at  $[\zeta, \varsigma] = [10^{-3}; 0.02]^T$  while keeping the hyperparameters at the same value. The associated residual and decision function are displayed in Figure 9. The residual reacts more strongly to this more severe fault than in the worst case, and will therefore lead to easier decision. The decision function indeed raises an alarm very shortly after the occurrence of the fault. To illustrate more precisely the performance obtained with the worst-case optimal tuning of the hyperparameters on the entire environmental space, Figure 10 shows the value of the objective function  $y$  over  $\mathbb{X}_e$ .

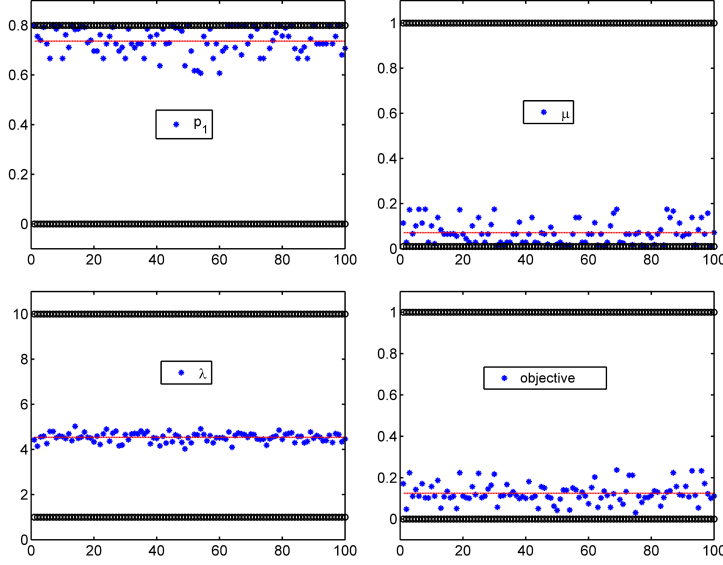


Fig. 7: Dispersion for hyperparameters  $p_1, \mu, \lambda$  and the objective function. Red line indicates the mean value and thick black lines correspond to space boundaries

## 6 Conclusions and perspectives

A new strategy to deal with continuous minimax optimization for costly-to-evaluate black-box functions has been described in this paper. With [27], this is one of the first attempt reported to address this difficult problem. The main loop of the algorithm uses a relaxation procedure to facilitate the search for an approximate solution. Kriging-based optimization deals with the fact that the objective function is expensive to evaluate. EGO has been employed for this purpose, but other strategies such as those described in [3, 9] may be used instead without the need to modify the entire algorithm.

The procedure has been tested on two continuous minimax benchmarks with known results taken as references, and reasonably accurate values of the minimax optimum have been obtained for any initialization of the algorithm. A simplified

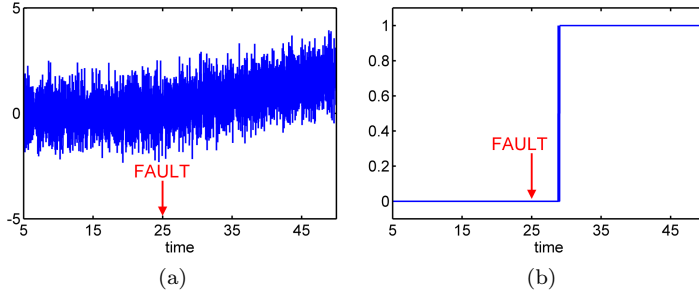


Fig. 8: Residual (a) and Boolean decision (b) for the mean of the estimated worst-case environmental variables  $\mathbf{x}_e = [9.3 \cdot 10^{-4}; 1.1 \cdot 10^{-3}]^T$ , with the mean of the estimates of the minimax-optimal hyperparameters

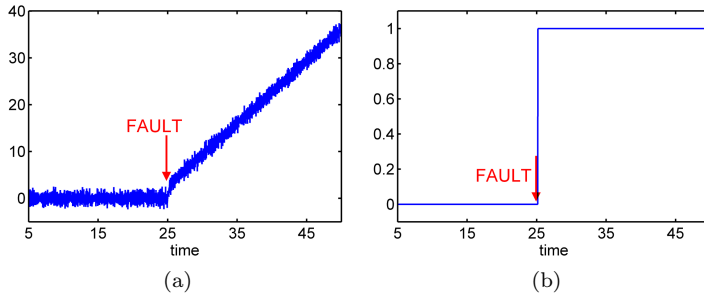


Fig. 9: Residual (a) and Boolean decision (b) for a randomly chosen value of the environmental variables  $\mathbf{x}_e = [10^{-3}; 0.02]$ , with the mean of the estimates of the minimax-optimal hyperparameters

academic version of a practical application in fault diagnosis has also been addressed and the results obtained are consistent with what common sense suggests. In both cases, relatively few evaluations of the black-box objective function have been needed, which is very interesting for costly applications where such designs are in high demand.

The only constraint on the control and environmental variables considered here was that they belonged to simple-shaped compact sets. The relaxation procedure should be able to deal with more complicated sets and constraints, even though this remains to be assessed through future tests. Building on these initial successful results, higher-dimensional practical problems in fault detection and isolation for aerospace vehicles are under study.

**Acknowledgements** The authors would like to thank Frédéric Damongeot (ONERA DCPS) for valuable discussions and comments, and Dr Michael Sasena for having made available the software SuperEGO, including the DIRECT algorithm, to the research community.

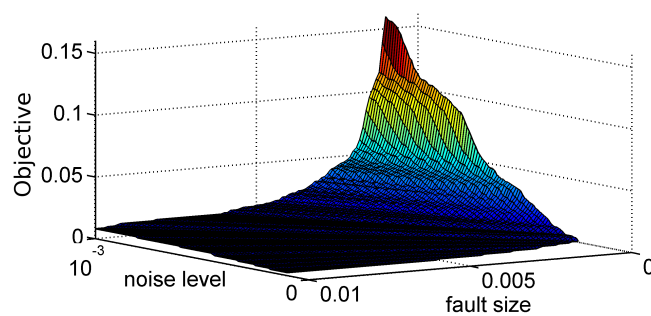


Fig. 10: Value of the objective function over  $\mathbb{X}_e$  for the mean of the estimates of the minimax-optimal hyperparameters

## References

1. T. J. Santner, B. J. Williams, and W. Notz. *The Design and Analysis of Computer Experiments*. Springer-Verlag, Berlin-Heidelberg, 2003.
2. D.R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21(4):345–383, 2001.
3. N. V. Queipo, R.T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, and P. K. Tucker. Surrogate-based analysis and optimization. *Progress in Aerospace Sciences*, 41(1):1–28, 2005.
4. T. W. Simpson, J. D. Poplinski, P. N. Koch., and J. K. Allen. Metamodels for computer-based engineering design: survey and recommendations. *Engineering with Computers*, 17(2):129–150, 2001.
5. M. D. McKay, R. J. Beckman, and W. J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245, 1979.
6. G. Matheron. Principles of geostatistics. *Economic Geology*, 58(8):1246–1266, 1963.
7. C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning*. Springer-Verlag, New York, 2006.
8. D. R. Jones, M. J. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.
9. A. I. J. Forrester, A. Sobester, and A. J. Keane. *Engineering Design via Surrogate Modelling: a Practical Guide*. Wiley, Chichester, 2008.
10. D. Huang, T. T. Allen, W. I. Notz, and N. Zeng. Global optimization of stochastic black-box systems via sequential kriging meta-models. *Journal of Global Optimization*, 34(3):441–466, 2006.
11. M.J. Sasena. *Flexibility and Efficiency Enhancements for Constrained Global Design Optimization with Kriging Approximations*. PhD thesis, University of Michigan, USA, 2002.
12. J. Villemonteix, E. Vazquez, and E. Walter. An informational approach to the global optimization of expensive-to-evaluate functions. *Journal of Global Optimization*, 44(4):509–534, 2009.
13. E. Vazquez and J. Bect. Convergence properties of the expected improvement algorithm with fixed mean and covariance functions. *Journal of Statistical Planning and Inference*, 140(11):3088–3095, 2010.
14. D. Huang and T.T. Allen. Design and analysis of variable fidelity experimentation applied to engine valve heat treatment process design. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 54(2):443–463, 2005.
15. J. Villemonteix, E. Vazquez, and E. Walter. Bayesian optimization for parameter identification on a small simulation budget. In *Proceedings of the 15th IFAC Symposium on System Identification, SYSID 2009, Saint-Malo France*, 2009.

16. J. Marzat, E. Walter, H. Piet-Lahanier, and F. Damongeot. Automatic tuning via Kriging-based optimization of methods for fault detection and isolation. In *Proceedings of the IEEE Conference on Control and Fault-Tolerant Systems, SYSTOL 2010, Nice, France*, pages 505–510, 2010.
17. J. Defretin, J. Marzat, and H. Piet-Lahanier. Learning viewpoint planning in active recognition on a small sampling budget: a Kriging approach. In *Proceedings of the 9th IEEE International Conference on Machine Learning and Applications, ICMLA 2010, Washington, USA*, pages 169–174, 2010.
18. H. G. Beyer and B. Sendhoff. Robust optimization – a comprehensive survey. *Computer Methods in Applied Mechanics and Engineering*, 196(33-34):3190–3218, 2007.
19. G. Dellino, J. P. C. Kleijnen, and C. Meloni. Robust optimization in simulation: Taguchi and response surface methodology. *International Journal of Production Economics*, 125(1):52–59, 2010.
20. W. Chen, J. K. Allen, K. L. Tsui, and F. Mistree. A procedure for robust design: minimizing variations caused by noise factors and control factors. *ASME Journal of Mechanical Design*, 118:478–485, 1996.
21. K. Lee, G. Park, and W. Joo. A global robust optimization using the kriging based approximation model. In *Proceedings of the 6th World Congresses of Structural and Multidisciplinary Optimization, Rio de Janeiro, Brazil*, 2005.
22. B. J. Williams, T. J. Santner, and W. I. Notz. Sequential design of computer experiments to minimize integrated response functions. *Statistica Sinica*, 10(4):1133–1152, 2000.
23. J. S. Lehman, T. J. Santner, and W. I. Notz. Designing computer experiments to determine robust control variables. *Statistica Sinica*, 14(2):571–590, 2004.
24. C. Q. Lam. *Sequential Adaptive Designs in Computer Experiments for Response Surface Model Fit*. PhD thesis, The Ohio State University, 2008.
25. A. M. Cramer, S. D. Sudhoff, and E. L. Zivi. Evolutionary algorithms for minimax problems in robust design. *IEEE Transactions on Evolutionary Computation*, 13(2):444–453, 2009.
26. R. I. Lung and D. Dumitrescu. A new evolutionary approach to minimax problems. In *Proceedings of the 2011 IEEE Congress on Evolutionary Computation, New Orleans, USA*, pages 1902–1905, 2011.
27. A. Zhou and Q. Zhang. A surrogate-assisted evolutionary algorithm for minimax optimization. In *Proceedings of the 2010 IEEE Congress on Evolutionary Computation, Barcelona, Spain*, pages 1–7, 2010.
28. K. Shimizu and E. Aiyoshi. Necessary conditions for min-max problems and algorithms by a relaxation procedure. *IEEE Transactions on Automatic Control*, 25(1):62–66, 1980.
29. B. Rustem and M. Howe. *Algorithms for Worst-Case Design and Applications to Risk Management*. Princeton University Press, 2002.
30. B. Brown and T. Singh. Minimax design of vibration absorbers for linear damped systems. *Journal of Sound and Vibration*, 330(11):2437–2448, 2011.
31. D. M. Salmon. Minimax controller design. *IEEE Transactions on Automatic Control*, 13(4):369–376, 1968.
32. J. Helton. Worst case analysis in the frequency domain: The  $H_\infty$  approach to control. *IEEE Transactions on Automatic Control*, 30(12):1154–1170, 1985.
33. E. Y. Chow and A.S. Willsky. Analytical redundancy and the design of robust failure detection systems. *IEEE Transactions on Automatic Control*, 29:603–614, 1984.
34. P. M. Frank and X. Ding. Survey of robust residual generation and evaluation methods in observer-based fault detection systems. *Journal of Process Control*, 7(6):403–424, 1997.
35. B. Colson, P. Marcotte, and G. Savard. An overview of bilevel optimization. *Annals of Operations Research*, 153(1):235–256, 2007.
36. A. Ben-Tal and A. Nemirovski. Robust convex optimization. *Mathematics of Operations Research*, 23(4):769–805, 1998.
37. T. Başar and G. J. Olsder. *Dynamic Noncooperative Game Theory*. Society for Industrial Mathematics, New York, 1999.
38. D. Du and P. M. Pardalos. *Minimax and Applications*. Kluwer Academic Publishers, Norwell, 1995.
39. P. Pappas and B. Rustem. An algorithm for the global optimization of a class of continuous minimax problems. *Journal of Optimization Theory and Applications*, 141(2):461–473, 2009.
40. A. Tsoukalas, B. Rustem, and E. N. Pistikopoulos. A global optimization algorithm for generalized semi-infinite, continuous minimax with coupled constraints and bi-level problems. *Journal of Global Optimization*, 44(2):235–250, 2009.



41. B. Rustem. *Algorithms for Nonlinear Programming and Multiple Objective Decisions*. John Wiley & Sons, Chichester, 1998.
42. K. Shimizu, Y. Ishizuka, and J. F. Bard. *Nondifferentiable and Two-level Mathematical Programming*. Kluwer Academic Publishers, Norwell, 1997.
43. D. J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge Univ Press, 2003.
44. M. Schonlau. *Computer Experiments and Global Optimization*. PhD thesis, University of Waterloo, Canada, 1997.
45. D. R. Jones, C. D. Perttunen, and B. E. Stuckman. Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Applications*, 79(1):157–181, 1993.
46. M. Basseville and I. V. Nikiforov. *Detection of Abrupt Changes: Theory and Application*. Prentice Hall, Englewood Cliffs, 1993.
47. M. Bartys, R. J. Patton, M. Syfert, S. de las Heras, and J. Quevedo. Introduction to the DAMADICS actuator FDI benchmark study. *Control Engineering Practice*, 14(6):577–596, 2006.